Fast, Approximate Algorithm for Detection of Solvent-Inaccessible Atoms

JÖRG WEISER, PETER S. SHENKIN, W. CLARK STILL
Department of Chemistry, Columbia University, New York, New York 10027

Received 21 July 1998; accepted 11 November 1998

ABSTRACT: Up to about half of the atoms in biopolymers are inaccessible to solvents. If such atoms can be rapidly identified, time can be saved in the subsequent computation of atomic surface areas. A quick, approximate method, termed buried atom elimination (BAE), was developed for the detection of such atoms. Following the literature, the method makes use of a Gaussian function to calculate the neighbor density in four tetrahedral directions in 3-dimensional space, sometimes twice with different orientations. In macromolecules, our method detects between 63 and 81% of the buried atoms but also incorrectly classifies 2–8% of the exposed atoms as buried. These misidentified atoms all have small solvent-exposed (accessible) surface areas (SASAs): their surfaces sum to a maximum of 0.5% of the molecular SASA, and their maximum atomic SASA is 5.1 Å^2. Using our recently reported LCPO method for computing atomic surfaces, which is one of the fastest available, the use of BAE increases the overall speed of computing the atomic SASAs by a factor of up to 1.6 for surfaces only and 1.9 when first and second derivatives are computed. BAE decreases the LCPO average absolute atomic error from about 2.3 Å^2 to about 1.7 Å^2 (average for larger compounds). BAE was introduced into the MacroModel molecular modeling package and tests show that it increases the efficiency of first- and second-derivative energy minimizations and molecular dynamics simulations without adversely affecting the stability or accuracy of the calculations. BAE parameters were developed for the most important atom types in biopolymers, based on a parameterization set of 18 compounds of different size (33–4346 atoms) and class (organics, proteins, DNA, and various complexes), consisting of a total of 23,186 atoms. J Comput Chem 20, 586–596, 1999

Keywords: buried atoms; solvent-accessible surface area; analytical surface areas; derivatives; neighbor-list reduction

Address Correspondence to: J. Weiser; e-mail: joerg@still3.chem.columbia.edu

Contract/grant sponsor: Deutsche Forschungsgemeinschaft
Contract/grant sponsor: NSF; contract/grant number: CHE97-07870

CCC 0192-8651 / 99 / 060586-11

Introduction

The solvent-accessible surface area (SASA) of a molecule is widely used in describing solvation of solutes and macromolecules. The definition of the SASA was given by Lee and Richards,[1] who presented the image of rolling a sphere, representing a solvent molecule, over the van der Waals surface (vdWSA) of a protein. The SASA is described by the locus of points swept out by the center of the solvent sphere. Much of the current interest in the SASA is due to the observation that, at least for nonpolar molecules, the free energy of aqueous solvation is roughly proportional to this quantity,[2] which in turn is roughly proportional to the number of solvent molecules that can contact the solute molecule. Many programs calculate analytical atomic SASAs and their derivatives with respect to atomic coordinates.[3–20]

SASA may be thought of as the vdWSA of a molecule in which all the atoms' radii have been increased by the radius of the solvent probe. This increase in effective atomic radius causes the distribution of atomic surface areas to differ greatly for these two types of surfaces. Figure 1 shows the distribution of united-atom atomic vdWSAs in 18 compounds of different size (33–4346 atoms) and class (small organics, proteins, DNA, and various complexes) consisting of 23,186 atoms (Table I). Figure 2 shows the distribution of the united-atom atomic SASAs for the same data set, using a solvent radius of 1.4 Å. The computations were performed in MacroModel/BatchMin[21] using a numerical method with atomic radii similar to those used in the OPLSA force field.[22] Only one atom in the data set has a vdWSA of 0.0 Å^2, and the maximum atomic vdWSA is 43.6 Å^2. For SASA, 9367 (40.4%) of the atoms are completely inaccessible to the solvent and 12,254 atoms (52.9%) have SASA values smaller than 1.0 Å^2. The largest SASA observed is 75.7 Å^2. Thus, nearly half of the atoms have no or almost no contribution to the total SASA.

If it were possible to detect some or all of the buried atoms in a rapid preprocessing step, one would expect subsequent surface calculations to proceed nearly 2 times faster. Moreover, the first and second derivatives of SASA with respect to atomic coordinates, where needed, could be set to zero for these atoms, conferring at least corresponding savings when computing these quantities. A buried-atom elimination (BAE) procedure does not have to be mathematically exact. Atoms that are in fact buried but are not identified by BAE are not problematic, because here we suffer only from incomplete optimization. Atoms that are

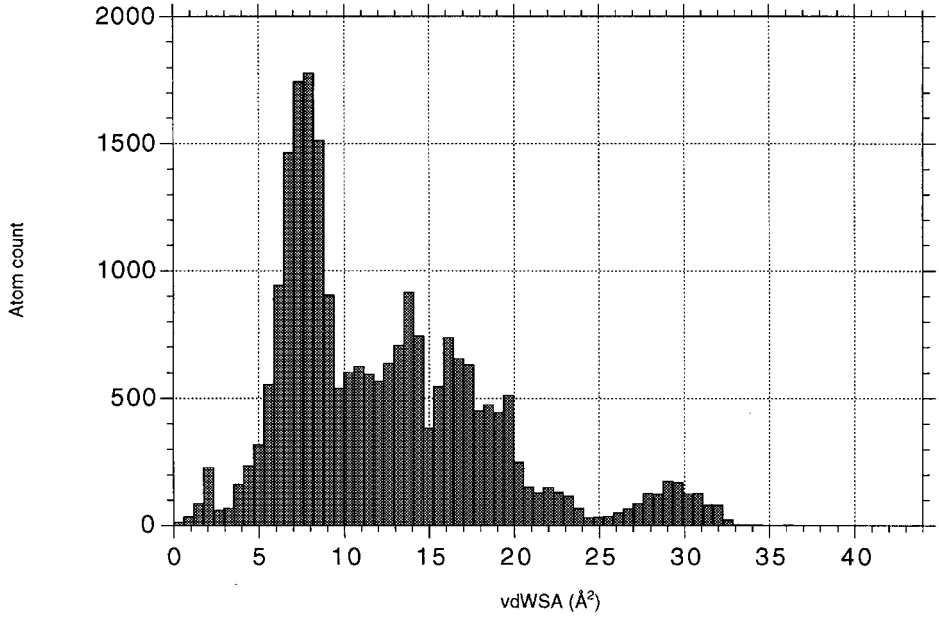

FIGURE 1. Atomic distribution, vdWSA. Distribution of 23,186 atomic vdWSAs of the 18 test compounds from Table I.

TABLE I.
Compounds

ctc	7-Chlorotetracycline ($C_{22}H_{23}N_2O_8Cl$); complexed with tetracycline repressor from *Escherichia coli* (Brookhaven entry 2tct)
sip	Sipholenol-a monoacetate ($C_{32}H_{54}O_5$); global MM3(92) energy minimum[29]
nmx	Nitromethyldethia coenzyme A ($C_{22}H_{37}N_8O_{18}P_3$); complexed with chicken citrate synthase complex (Brookhaven entry 1amz); two conformers of side chain given in pdb file; we took conformer A
1van	Vancomycin complex with L-Lys-D-Ala-D-ala (theoretical model)
1crn	Crambin
103d	DNA (5′-D(*GP*TP*GP*GP*AP*AP*TP*GP*GP*AP*AP*C)-3′) (antiparallel DNA duplex; human centromere repeat)
2ins	Insulin from bovine (*Bos taurus*)
163d	Rev responsive element (RBE, 30 ribonucleotide fragment) complexed with HIV rev protein (residues 34–50)
1lz1	Human lysozyme
2stw	Human ETS1 / DNA complex
2tra	Transfer ribonucleic acid (yeast, Asp) with spermine ($C_{10}H_{26}N_4$); we took conformer A
1sbg	HIV-1 protease complexed with the inhibitor SB203386
5tra	Transfer ribonucleic acid (yeast); metal atom in pbd file (M7, atom 255) not taken into account
1inc	Porcine pancreatic elastase complex with benzoxazinone inhibitor ($C_{17}H_{22}N_2O_4Cl$)
1kvd	Killer toxin from halotolerant yeast
3app	Fungus acid proteinase (penicillopepsin)
1bni	Microbial ribonuclease (barnase wild-type structure at pH 6.0)
1ca0	Bovine chymotrypsin complexed to inhibitor domain of Alzheimer amyloid

All acronyms containing four characters are Brookhaven entries.[28]

not buried but are identified as buried are potentially more problematic; but if such errors can be made small, the approximation may be acceptable. Furthermore, because most analytical SASA calculations deviate somewhat from exact SASA results, small BAE errors in which incorrectly identified buried atoms have only small SASAs may not significantly worsen the overall approximation.

The simplest situation in which an atom might be buried would be that it is entirely contained

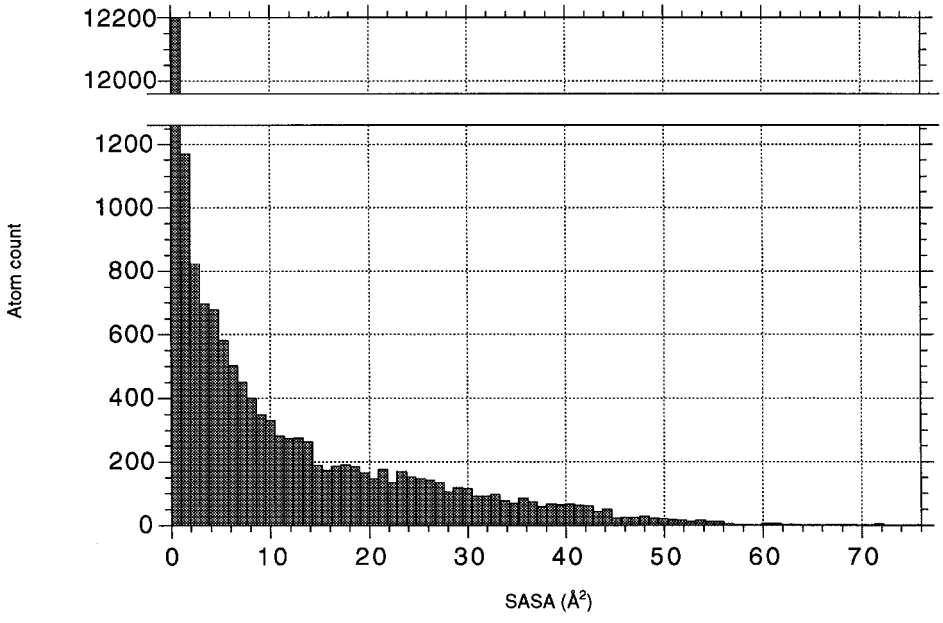

FIGURE 2. Atomic distribution, SASA. Distribution of 23,186 atomic SASAs of the 18 test compounds from Table I.

within another sphere, as described by Richmond[4] and Gibson and Scheraga.[23] Such a situation did not occur in our data set. Gogonea and Osawa[24] described an algorithm to detect a sphere fully buried within the union of two neighboring spheres. We did not encounter this situation either, in our test set with our radii. It is possible that such examples might be found where explicit hydrogens are used; however, our data set is a united-atom data set, which is consistent with the way in which SASAs are computed in MacroModel. In our data set no atom is completely buried by fewer than four overlapping neighbors: all such examples are sp^3 carbons with four bonded neighbors. We found that some buried atoms require as many as 10 neighbors to completely bury them. It does seem not worthwhile to attempt to extend the approach of Gogonea and Osawa[24] to a greater number of neighboring spheres, because such a process would likely demand more CPU time than simply calculating the SASAs. Fraczkiewicz and Braun[18] presented a method for detecting buried atoms by calculating the intersection of half-spaces; however, this procedure is embedded within a complete surface area algorithm. Unfortunately, for our purpose, they did not provide information concerning the CPU cost for the subprocedure.

Stouten et al.[25] published a method for computing the ''neighbor density'' (ND) of atoms in molecules. This is a weighted sum of the number of neighbors of a given atom in which the weighting is given by a Gaussian function centered on the atom. ND is used in solvation studies in a manner similar to the way in which SASA is used: the solvation energy is computed by multiplying the difference between an atom's maximum and current ND by a constant characteristic of atom type. Thus, to the extent to which ND is a good measure for this purpose, it should be at least roughly proportional to SASA (with a negative constant of proportionality).

Stouten computes a single ND for each atom without distinguishing the relative positions of the neighbors. Thus, the method may have trouble distinguishing between a loose but spherically symmetric set of neighbors that suffices to bury the central atom and a tight but one-sided distribution that does not. The method described here is a direct elaboration of Stouten's method. Rather than consider a single ND for each atom, we found it beneficial to consider four separate NDs oriented in tetrahedral directions. Only when all four NDs were sufficiently great was the atom regarded as buried. Four tetrahedral directions were chosen because this is the minimum number sufficient to symmetrically span 3-dimensional space.

Method

Given a candidate atom for BAE, four tetrahedral rays are defined as follows. The rays start at the central atom and point in the directions of the vertices of a regular tetrahedron centered on the atom. The first ray points toward the atom's nearest neighbor; the second is as close as possible to the second nearest neighbor and the other two are then uniquely determined. In computing the four NDs, each neighboring atom is assigned to the ray to which it lies closest. Conceptually, each ray forms the axis of a tetrahedral ''cone'' containing the atoms contributing to its ND.

The four tetrahedral rays t_1–t_4 of atom i (the central atom) are represented by four unit vectors $\underline{n}_{t_1}$ through $\underline{n}_{t_4}$. N_1 and N_2 are the closest and second closest atomic neighbors to the central atom, and $\underline{n}_{N_1}$ is a unit vector pointing toward N_1. $\underline{n}_{t_1}$ is then defined as follows:

$$\underline{n}_{t_1} = \underline{n}_{N_1}. \tag{1}$$

Because the dot product of two tetrahedral unit vectors is given by

$$\underline{n}_{t_1} \cdot \underline{n}_{t_2} = -\frac{1}{3}, \tag{2}$$

the following equation defines t_2 (Fig. 3):

$$\underline{n}_{t_2} = -\frac{1}{3}\underline{n}_{t_1} + \sqrt{\frac{8}{9}}\,\underline{n}_b. \tag{3}$$

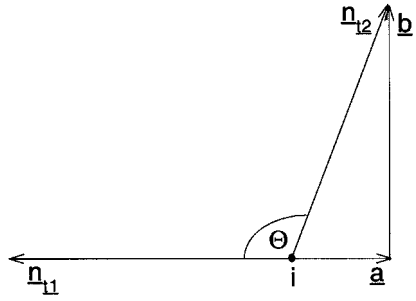

FIGURE 3. Definition of the second tetrahedral ray t_2. When Θ is the tetrahedral angle, $|\underline{a}| = -\cos\Theta = 1/3$. Because $|\underline{a}|^2 + |\underline{b}|^2 = 1$, we have $|\underline{b}| = \sqrt{8/9}$. All points and vectors shown are in the N_1, i, N_2 plane.

$\underline{n}_b$ is perpendicular to t_1. We require t_2 to be as close as possible to N_2; if $\underline{N}_2$ is the vector from i to N_2, then $\underline{b}$ and $\underline{n}_b$ are given by

$$\underline{b} = (\underline{n}_{t_1} \times \underline{N}_2) \times \underline{n}_{t_1}, \quad (4)$$

$$\underline{n}_b = \frac{\underline{b}}{|\underline{b}|}. \quad (5)$$

If $\underline{n}_{t_1}$ and $\underline{N}_2$ are parallel to each other or nearly so, as in alkynes, then the parenthetical cross product in eq. (4) will be close to zero and the overall equation is ill conditioned. This is checked using the criterion shown in eq. (6), which implies an angle within about 3° of linearity.

$$|\underline{n}_{t_1} \times \underline{N}_2| < 0.05. \quad (6)$$

If this criterion is met, we replace this cross product by an arbitrarily chosen vector not parallel to $\underline{n}_{t_1}$ and proceed with eq. (4). [In practice, we use the vector (1, 0, 0) unless this also satisfies eq. (6), in which case we use (0, 1, 0).]

The values t_3 and t_4 are set by two rotations[26] of t_1 and t_2. First, a rotation of 180° about $\underline{c}$, the axis connecting $\underline{n}_{t_1}$ and $\underline{n}_{t_2}$, takes these points to $\underline{n}_{t_5}$ and $\underline{n}_{t_6}$, respectively.

$$\underline{c} = \underline{n}_{t_1} - \underline{n}_{t_2}. \quad (7)$$

A second rotation of 90° about $\underline{d}$, which bisects $\underline{n}_{t_5}$ and $\underline{n}_{t_6}$, transforms these points into $\underline{n}_{t_3}$ and $\underline{n}_{t_4}$, respectively.

$$\underline{d} = \underline{n}_{t_5} + \underline{n}_{t_6}. \quad (8)$$

The neighbor density $\rho_{i,k}$ for tetrahedral direction k of atom i is calculated as

$$\rho_{i,k} = \sum_j^{m_{i,k}} \exp\left(-\alpha_i \cdot \frac{d_{ij}^2}{(r_{H_2O} + r_j)^2}\right), \quad (9)$$

where $m_{i,k}$ is the number of neighbors of i in direction k, α_i is a constant that depends on the atom type of i, d_{ij} is the distance between atoms i and j, r_j is the vdW radius of neighboring atom j, and r_{H_2O} is the probe size of water (1.4 Å). An atom is defined as buried if all four tetrahedral neighbor densities $\rho_{i,k}$ are above a certain limit ρ_i^*, which is characteristic of the atom type. The goal in determining an appropriate value for ρ_i^* is to detect as many buried atoms as possible without defining as buried atoms those that are highly exposed.

The calculated neighbor densities $\rho_{i,k}$ are based on all overlapping neighbors of atom i. Attempts at lowering the number of neighbors by removing remote neighbors from the neighbor list of i lowers the CPU time but also reduces accuracy in detecting buried atoms. Neighbor-list reduction (NLR),[27] a procedure that removes atoms whose elimination cannot affect the exposed surface of i and that considerably shortens the neighbor list when SASA is being computed, diminishes the accuracy of BAE, even though NLR is exact for surfaces. Therefore, we do not use NLR for BAE, even though we do use it for computing SASAs.

Parameters α_i and ρ_i^* were developed for different atom types, which are based on atomic number, hybridization, and number of bonded neighbors.[20] Two different algorithms for BAE are distinguished: single and double check of $\rho_{i,k}$. If an sp^3 atom has four bonded neighbors, t_1–t_4 point rather closely toward the four bonded neighbors. Here the neighbor density is checked only once. If one considers an atom that has only one bonded neighbor (e.g., oxygen in carbonyl), the first tetrahedral ray points toward the bonded carbon atom; but the other three tetrahedrals point in arbitrary directions, depending on the closest nonbonded neighbor of the oxygen. Here if the first tetrahedral orientation defines the atom as buried, we check the NDs in a second tetrahedral orientation: the three tetrahedral rays t_2–t_4 are rotated 180° about the axis of t_1 and the neighbor densities $\rho_{i,k}$ are recalculated. The atom is defined as buried if all eight $\rho_{i,k}$ are above ρ_i^*. This double check leads to higher accuracy in BAE for atom types that are not intrinsically tetrahedral.

The parameterization of α_i and ρ_i^* is based on the compounds listed in Table I. Parameters were determined only for those atom types for which 10% or more of the atoms are buried. In addition, we did not determine parameters of any atom type for which there were 10 or fewer buried atoms. The smaller the value of ρ_i^*, the more buried atoms are identified. On the other hand, lower values of ρ_i^* also identify more atoms as buried that are in fact exposed. Because we set the derivatives of buried atoms to zero, we have to make sure that the maximum atomic error does not become too large in order to avoid large discrepancies in derivatives when using this approach in minimization or molecular dynamics (MD). Therefore, α_i and ρ_i^* were chosen such that the maximum atomic error was not larger than about 5 Å and the number of correctly detected buried atoms was greatest; subject to this, minimizing the num-

ber of incorrectly identified exposed atoms was used as a secondary criterion.

Results and Conclusions

We calculated the SASAs (Table II) of the 18 compounds listed in Table I. The data set consisted of 23,186 atoms. We used atomic radii similar to OPLSA vdW radii[22] as implemented in MacroModel[21] and a solvent-probe radius of 1.4 Å. All coordinates for the data set are published in refs. 28 and 29. Numerical SAs required for the parameterization were computed using MacroModel/BatchMin.[21] Based on these results, parameters for the approximate detection of solvent-inaccessible atoms were derived (Table III). This method (BAE) was applied as a preprocessing step to the LCPO method of computing analytical atomic SASAs and first and second derivatives with respect to Cartesian coordinates.[20] The CPU times of BAE and LCPO with and without BAE are given in Table IV. In this table the results shown for first derivatives include times for the computation of surfaces as well; times shown for second derivatives include those for surfaces and first derivatives. The times shown for SASA calculations with BAE list the BAE overhead separately. None of the times include the construction of the pairwise interatomic distance matrix that is used to construct the neighbor list, because this is common to all the calculations. CPU times for construction of the distance matrix can be found in ref.[27].

All calculations were performed on a SGI R10000/194 MHz processor (Power Onyx), using Fortran code optimized at the -n32 -mips3 -O3 level.

Table III shows all atom types occurring in our test set and their BAE parameters, where derived. All sp^3 atoms with more than one bonded neighbor yield best BAE results when the single-check method is performed; in all other cases, double checking is performed, so that tetrahedral neighbor densities are calculated twice for such central atoms. In the case of carbon sp^3 atoms with four bonded neighbors, no BAE parameters were developed, because all such atoms in the test set were

TABLE II.
BAE Results

Compounds	No Atoms	No. Buried Atoms (% of All Atoms)	No Correctly Detected BAE Atoms (% of all Buried Atoms)	No. Incorrectly Detected BAE Atoms (% of All Exposed Atoms)	Ave. Abs. Atomic BAE Error (Å^2)[a]	Ave. Abs. Exposed Atomic Error (Å^2)[b]	Max. Abs. Atomic Error (Å^2)[c]	Total SASA Error (Å^2) (% of Total SASA)	Total SASA (Å^2)	Ave. Abs. Atomic Error (Å^2) with and without BAE
ctc	33	2(6)	2(100)	0(0)	0.00	0.0	0.0	0.0(0.0)	623.8	2.96 / 2.96
sip	37	7(19)	5(71)	0(0)	0.00	0.0	0.0	0.0(0.0)	762.1	3.93 / 3.93
nmx	51	2(4)	1(50)	1(2)	0.44	0.9	0.9	0.9(0.1)	903.9	2.90 / 2.90
1van	121	21(17)	16(76)	6(6)	0.17	0.6	1.8	3.7(0.2)	1490.7	3.43 / 2.90
1crn	327	104(32)	73(70)	5(2)	0.04	0.6	1.3	3.0(0.1)	3010.9	2.49 / 2.29
103d	500	144(29)	116(81)	22(6)	0.17	1.1	3.8	23.6(0.5)	4338.4	2.40 / 2.16
2ins	770	304(39)	195(64)	18(4)	0.07	0.8	4.0	14.6(0.3)	5626.8	2.52 / 2.18
163d	813	285(35)	221(78)	40(8)	0.10	0.7	3.3	27.3(0.5)	5554.1	2.42 / 2.06
1lz1	1029	432(42)	316(73)	44(7)	0.08	0.6	4.2	27.8(0.4)	6624.2	2.33 / 1.83
2stw	1488	348(23)	219(63)	67(6)	0.18	0.8	4.2	51.1(0.4)	11,785.3	2.73 / 2.57
2tra	1544	465(30)	345(74)	83(8)	0.12	0.6	4.1	52.9(0.4)	12,344.4	2.55 / 2.22
1sbg	1559	687(44)	436(63)	57(7)	0.05	0.4	2.4	25.3(0.3)	9410.0	2.36 / 1.92
5tra	1821	514(28)	351(68)	93(7)	0.13	0.6	3.1	58.6(0.4)	14,675.9	2.68 / 2.31
1inc	1846	868(47)	628(72)	57(6)	0.02	0.4	2.5	22.1(0.2)	10,414.7	2.09 / 1.62
1kvd	1988	970(49)	697(72)	43(4)	0.03	0.5	3.1	21.2(0.2)	10,807.8	2.32 / 1.76
3app	2366	1145(48)	825(72)	92(8)	0.04	0.4	3.3	37.7(0.3)	12,508.2	2.23 / 1.70
1bni	2547	1045(41)	777(74)	95(6)	0.05	0.5	5.1	44.7(0.3)	16,812.7	2.32 / 1.86
1ca0	4346	2024(47)	1524(75)	177(8)	0.06	0.6	4.1	97.5(0.4)	24,336.4	2.14 / 1.65

[a] Calculated as follows: SASA of all BAE atoms / number of all BAE atoms.
[b] Calculated as follows: SASA of all BAE atoms / number of exposed BAE atoms.
[c] Atomic surfaces range from 0.0 to 75.7 Å^2.

TABLE III.
BAE Parameter.

Atom Type, No. of Bonded Neighbors	No. Atoms in Data Set[a]	No. Buried Atoms in Data Set (%)[a]	α_i[b]	ρ_i^*[b]	Reorientation of Tetrahedrals?	Max. Atomic Error (Å)[f]	Ave. Abs. Atomic Error (Å)[g]
C *sp*3, 1	1360	532(39)	0.72	1.59	Yes	4.1	0.11
C *sp*3, 2	3095	763(25)	0.73	2.11	No	3.2	0.10
C *sp*3, 3	3701	1773(48)	0.68	2.16	No	3.5	0.03
C *sp*3, 4	11	11(100)	[c]	[c]	[c]	[c]	[c]
C *sp*2, 2	1464	569(39)	0.65	2.08	Yes	4.2	0.25
C *sp*2, 3	4031	2284(57)	0.65	2.18	Yes	3.8	0.05
O *sp*3, 1	726	71(10)	0.79	1.62	Yes	0.8	0.05
O *sp*3, 2	732	148(20)	0.76	1.29	No	1.1	0.05
O *sp*2, 1	3253	1095(34)	0.52	2.37	Yes	5.1	0.08
O^- Carboxylate, 1	425	18(4)	[d]	[d]	[d]	[d]	[d]
N *sp*2, 1	594	48(8)	[d]	[d]	[d]	[d]	[d]
N *sp*2, 2	2960	1708(58)	0.81	1.07	Yes	2.5	0.04
N *sp*2, 3	320	271(85)	0.50	0.65	Yes	0.9	0.03
N *sp*3, 1	136	4(3)	[d]	[d]	[d]	[d]	[d]
N *sp*3, 2	20	3(15)	[e]	[e]	[e]	[e]	[e]
N *sp*3, 3	6	6(100)	[e]	[e]	[e]	[e]	[e]
S, 1	5	1(20)	[e]	[e]	[e]	[e]	[e]
S, 2	99	51(52)	0.74	1.99	No	1.5	0.08
P, 3	7	0(0)	[d]	[d]	[d]	[d]	[d]
P, 4	237	10(4)	[d]	[d]	[d]	[d]	[d]
Cl, 1	4	1(25)	[e]	[e]	[e]	[e]	[e]

[a] Eighteen test compounds from Table I using OPLSA vdW radii.[22]
[b] See eq. (9).
[c] LCPO parameters are zero; no BAE checking necessary.
[d] Less than 10% of atoms are buried; BAE is not performed.
[e] Not enough data for BAE parameter development.
[f] Atomic SASA range from 0.0 to 75.7 Å^2.
[g] Calculated as follows: SASA of all BAE atoms / number of BAE atoms.

completely buried. The parametrization yielded very good results: the average absolute atomic error, defined as the sum of the SASAs of all BAE atoms divided by their number, is between 0.03 and 0.11 Å^2 for all atom types except sp^2 carbon with two bonded neighbors, which shows an average absolute atomic error of 0.25 Å^2.

Table II shows the BAE results for all compounds based on the parameters shown in Table III. Roughly speaking, the fraction of buried atoms in a compound rises with the size of the compound and with the tendency of a molecule to adopt a spherical shape. Thus, proteins show a larger percentage of buried atoms than DNA/RNA, but larger proteins may not exhibit a higher fraction of buried atoms than small ones if they deviate significantly from sphericity. Our method correctly identifies between 63 and 81% of the buried atoms in macromolecules but also identifies 2–8% of the exposed atoms as buried; the average absolute atomic error of the atoms defined as buried, computed as described in the previous paragraph, ranges from 0.03 to 0.18 Å^2 for larger compounds and is about 0.05 Å^2 for the biggest compounds. The sum of the SASAs of the atoms identified as buried ranges from 0.0 to 0.5% of the total SASA for the compounds shown. The distribution of the SASAs of BAE atoms is shown in Figure 4, which shows a large peak at 0.0. The average actual SASA of the atoms identified as buried that are really exposed is about 0.5 Å^2 for most large compounds, but it is as large as 1.1 Å^2 in one case (103d).

Table IV provides CPU timings for the BAE method. Without taking into account the BAE overhead, pure surface and surface plus derivative calculations are both about 1.6 times faster for DNA/RNA and about 2.0 times faster for proteins. Of course, when discussing any actual SA computation method, BAE overhead should be taken

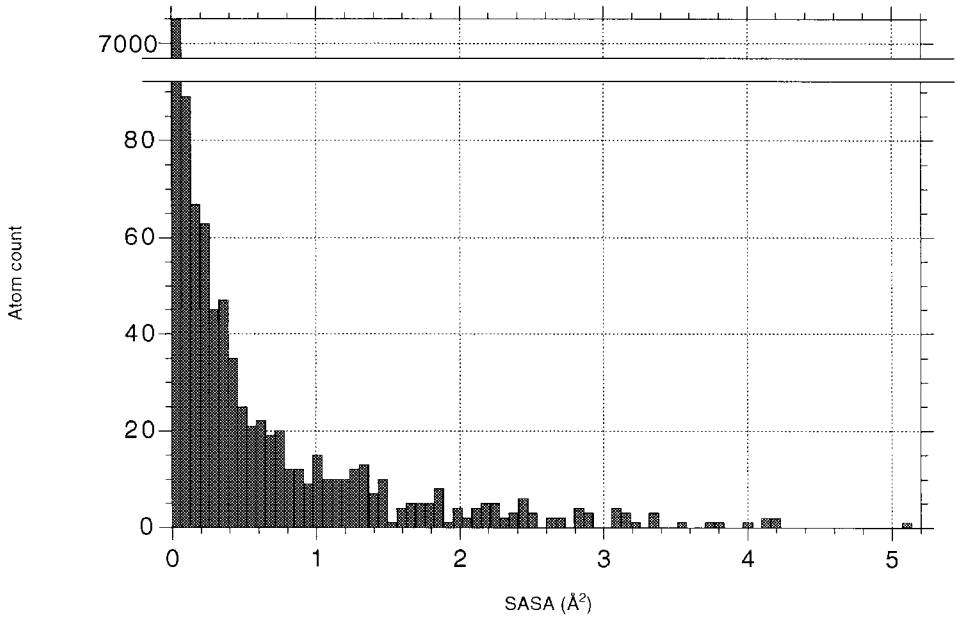

FIGURE 4. Atomic distribution, SASA / BAE. Distribution of the SASAs of BAE atoms of the 18 test compounds from Table I.

TABLE IV.
CPU Times for LCPO and LCPO _ BAE-SASA Calculations.

Compound	No. Atoms	CPU LCPO			CPU BAE Overhead (s)	CPU LCPO-BAE			LCPO Surface vs. LCPO-BAE Surface[d]	LCPO First Deriv. vs. LCPO-BAE First Deriv.[d]	LCPO Second Deriv. vs. LCPO-BAE Second Deriv.[d]
		Surface (s)	First Deriv. (s)	Second Deriv. (s)[b]		Surface (s)[c]	First Deriv. (s)[a,c]	Second Deriv. (s)[b,c]			
ctc	33	0.003	0.006	0.032	0.0006	0.003	0.006	0.029	0.83	0.91	1.08
sip	37	0.004	0.006	0.032	0.0007	0.003	0.005	0.027	1.08	1.05	1.16
nmx	51	0.005	0.009	0.048	0.001	0.005	0.009	0.045	0.83	0.90	1.04
1van	121	0.032	0.054	0.302	0.003	0.023	0.038	0.217	1.23	1.32	1.37
1crn	327	0.105	0.167	0.947	0.012	0.068	0.109	0.610	1.31	1.38	1.52
103d	500	0.185	0.298	1.710	0.019	0.106	0.173	1.002	1.48	1.55	1.67
2ins	770	0.271	0.420	2.349	0.031	0.164	0.256	1.438	1.39	1.46	1.60
163d	813	0.321	0.500	2.813	0.033	0.177	0.277	1.568	1.53	1.61	1.76
1lz1	1029	0.399	0.615	3.415	0.045	0.213	0.329	1.851	1.55	1.64	1.80
2stw	1488	0.463	0.727	3.962	0.054	0.340	0.535	2.900	1.18	1.23	1.34
2tra	1544	0.537	0.852	4.776	0.057	0.329	0.525	2.926	1.39	1.46	1.60
1sbg	1559	0.602	0.919	5.084	0.068	0.355	0.540	2.959	1.42	1.51	1.68
5tra	1821	0.632	1.011	5.658	0.065	0.418	0.672	3.754	1.31	1.37	1.48
1inc	1846	0.768	1.169	6.463	0.086	0.403	0.612	3.347	1.57	1.67	1.88
1kvd	1988	0.840	1.280	7.097	0.096	0.435	0.656	3.595	1.58	1.70	1.92
3app	2366	1.000	1.520	8.411	0.114	0.515	0.782	4.284	1.59	1.70	1.91
1bni	2547	0.984	1.521	8.421	0.114	0.523	0.814	4.487	1.54	1.64	1.83
1ca0	4346	1.881	2.829	[e]	0.212	0.943	1.406	[e]	1.63	1.75	[e]

[a] Calculation of first derivatives includes calculation of the surface.
[b] Calculation of second derivatives includes calculation of the surface and first derivatives.
[c] Not including BAE overhead.
[d] Calculated as follows: LCPO / (BAE + LCPO-BAE).
[e] Not enough memory to calculate second derivatives.

into account; however, doing so gives figures that depend strongly on the speed of the method, and slower methods give faster apparent speedups. The figures just given are useful in that they should be relatively independent of what subsequent surface computation method is used; they also afford insight into the performance of the algorithm.

The speedup of 2.0 for proteins is surprisingly high: taking bovine chymotrypsin (1ca0) as an example, the speedup is 2.0 for the surfaces alone. BAE identifies 35% of all its atoms as buried, so that 65% are identified as exposed. If CPU time depended only on the fraction of atoms exposed, we would expect a speedup of 1/0.65 (1.54). The reason for the observed superlinear effect lies in the fact that buried atoms generally have more overlapping neighbors than exposed atoms; therefore, the cost of calculating their surfaces is greater than for computing the surface of an exposed atom, which has a shorter neighbor list.

As a method of reference, we chose our LCPO method to calculate SASA.[20] The cost of performing BAE is about 11% of the CPU time taken by the LCPO surface calculation when BAE is not performed. This ratio is constant over all sizes and classes of compounds studied. When BAE overhead is taken into account, the LCPO method, although one of the fastest reported,[20] becomes still faster: 1.6 times (surface only) to 1.9 times (surface, first and second derivatives) faster for proteins and 1.4 times (surface only) to 1.6 times (surface, first and second derivatives) faster in DNA/RNA. The higher the percentage of buried atoms in a compound, the greater the savings realized by the use of BAE; but the break-even point is surpassed for SASA second derivatives of even the smallest compounds. For the LCPO method, which is approximate, the use of BAE decreases the average absolute atomic error in SASA from about 2.3 Å^2 to about 1.7 Å^2 (Table II). Thus, BAE makes this method more accurate. When BAE is used, the total computed SASA of each compound stays about the same, because most atoms with a (small) negative LCPO SA[20] are detected as buried, which tends to increase the computed SASA when BAE is used, whereas some others with positive LCPO SASAs larger than zero are also identified as buried, tending to diminish the overall value. Reparametrization of LCPO is therefore not necessary; the same set of parameters may be used for pure LCPO and LCPO/BAE.

The BAE method, although mathematically inexact, provides a good mixture of accuracy, efficiency, and speed. It may be used as a preprocessing step for approximate analytical surface calculations including, but not limited to, LCPO. For example, the approximate methods MSEED[8] and SASAD[14] give results deviating from exact atomic SASAs by values, on average, in the range of 0.04–0.35 Å^2. MSEED's largest deviation has been reported as 43 Å^2.[18] Computation of atomic SASA and first derivatives of a protein with 2325 atoms on a platform very similar to ours takes 1.30 s for MSEED, 1.13 s for SASAD (4, 12), and 1.47 s as SASAD (4, 24) (a higher level of accuracy).[18] Use of BAE would probably speed up the methods by about the same factor as it speeds up the LCPO method, and BAE might slightly improve the accuracy of these methods.

The BAE method was implemented in MacroModel/BatchMin[21] along with LCPO.[20] We studied the influence on minimizations and MD simulations of this preprocessing step, together with a second, recently published optimization, termed NLR, which considerably decreases computation time of numerical[27] and analytical[20] surface areas. As test compounds we chose DNA (103d, 500 atoms) and insulin (2ins, 770 atoms). The force field used was the AMBER* united atom force field[30] and our generalized Born (GB/SA) solvation model[31] for water. GB/SA treats solvent as an analytical dielectric continuum that starts near the vdW surface of the solute and extends to infinity. The model includes both GB-based solvent polarization terms and SA solvent displacement terms, computed using the LCPO method.[20]

We performed two minimizations: truncated Newton conjugate gradient (TNCG),[32] which involves the use of second derivatives, and Pollack-Ribiere conjugate gradient (PRCG),[33] which involves only first derivatives. The number of iterations was restricted to make the results comparable. In addition, we carried out an MD simulation at 300K with a time step of 1.5 fs and an equilibration period of 10 ps using the same force field and solvent model. Each calculation was performed 3 times: in the first run the pure LCPO surface method was used; in the second, BAE was added as a preprocessing step; in the last computation, NLR was added to LCPO/BAE. Table V lists the final CPU time and gradients for the minimizations. Although the calculation of the surface energy makes only a very small contribution to the total solvation energy, and although the computation of surfaces is not a major part of the overall computing time, the use of the two preprocessing steps BAE and NLR visibly decreases the CPU

TABLE V.
MacroModel / BatchMin Calculations on DNA (103d, 500 Atoms) and Insulin (2ins, 770 atoms).

Method	Type	103d (CPU; Final Gradient)	2ins (CPU; Final Gradient)
TNCG minimization, 50 (103d) and 100 iterations (2ins)	LCPO	611.6 s, 0.007 kJ / mol Å	2876.7 s; 0.012 kJ / mol Å
0.1 kJ / mol $Å^2$ TNCG Hessian cutoff	LCPO / BAE	517.2 s; 0.001 kJ / mol Å	2508.9 s; 0.003 kJ / mol Å
	LCPO / BAE / NLR	445.6 s; 0.002 kJ / mol Å	2271.1 s; 0.001 kJ / mol Å
PRCG minimization, 1200 iterations	LCPO	735.2 s; 0.18 kJ / mol Å	2368.8 s; 0.49 kJ / mol Å
	LCPO / BAE	709.1 s; 0.10 kJ / mol Å	2296.6 s; 0.48 kJ / mol Å
	LCPO / BAE / NLR	686.8 s; 0.17 kJ / .mol Å	2247.0 s; 0.47 kJ / mol Å
MD, 10ps	LCPO	1470.2 s	4070.3 s
	LCPO / BAE	1382.8 s	3862.0 s
	LCPO / BAE / NLR	1290.7 s	3697.0 s

TABLE VI.
Van der Waals Radii Used for Computation of SASA.

Atom Type	α Atoms	vdW Radius	Atom Type	α Atoms	vdW Radius	Atom Type	α Atoms	vdW Radius
O3	H200	1.4800E00	CC	C300	1.9800E00	NA	0000	1.6250E00
O3	0000	1.4800E00	CC	0000	1.9000E00	NB	0000	1.6250E00
O2	0000	1.4800E00	CD	O200	1.8750E00	NC	C200	1.6250E00
OM	0000	1.4800E00	CD	0000	1.8750E00	NC	CD00	1.6250E00
OA	0000	1.5350E00	CE	0000	1.9000E00	NC	0000	1.6250E00
CA	0000	1.9250E00	CF	0000	1.9000E00	ND	0000	1.6250E00
CB	S100	1.9530E00	C1	0000	1.8250E00	NE	0000	1.6250E00
CB	O000	1.9530E00	C2	N400	1.8750E00	NF	0000	1.6250E00
CB	N000	1.9530E00	C2	N2N2	1.8750E00	NG	0000	1.6250E00
CB	C200	1.9530E00	C2	O200	1.8750E00	NH	0000	1.6250E00
CB	CD00	1.9530E00	C2	H100	1.8750E00	NI	0000	1.6250E00
CB	0000	1.9530E00	C2	0000	1.8750E00	S1	0000	1.9526E00
CC	S100	1.8750E00	C3	N000	1.9000E00	SA	0000	1.7750E00
CC	N000	1.8750E00	C3	O000	1.9000E00	PO	0000	1.8700E00
CC	O300	1.8750E00	C3	0000	1.9000E00	Cl	0000	2.2086E00
CC	C100	1.8750E00	N1	0000	1.6000E00			
CC	C200	1.9550E00	N2	0000	1.7063E00			
CC	CD00	1.9550E00	N3	0000	1.7063E00			
CC	CA00	1.9550E00	N4	0000	1.6250E00			
CC	CB00	1.9525E00	N5	0000	1.6250E00			

MacroModel / BatchMin uses the table vdW radii for the 18 compounds from Table I (00 = dummy).

time in all cases. The overall effect of BAE on first-derivative methods (PRCG minimization and MD) is small. However, BAE speeds up the second-derivative based TNCG minimization method by a factor of 1.2 for both compounds, and the use of BAE and NLR together speeds it up by factors of 1.4 and 1.3 for DNA and insulin, respectively, in comparison with pure LCPO. The final gradients are comparable to or lower than those obtained with pure LCPO.

Summary

Up to about half of the atoms in biopolymers are inaccessible to solvents. If these atoms can be rapidly identified, subsequent computation of atomic SASAs can be performed more quickly. The fast, approximate method described here for doing this, termed BAE, correctly identifies a large fraction of the buried atoms. Even in the context of an efficient SASA computation method, the use of BAE significantly reduces the overall time for the computation of atomic SASAs in all but the smallest molecules when the BAE overhead is included in the timings. However, BAE is not useful in the computation of vdWSAs, because virtually no atoms are buried in this situation, at least when united atoms are used.

BAE identifies some slightly exposed atoms as buried. This inaccuracy does not appear to lead to instability in energy minimizations or MD. The BAE parameters we derived are optimized for the atomic vdW radii used in MacroModel/BatchMin 6.5, which closely resemble OPLSA radii. The use of other radii will require reparameterization. BAE is turned on by default in MacroModel/BatchMin 6.5. The vdW radii used for the computation of SASA are available in Table VI.

References

1. Lee, B.; Richards, F. M. J Mol Biol 1971, 55, 379–400.
2. Hermann, R. B. J Phys Chem 1972, 76, 2754–2759.
3. Wodak, S. J.; Janin, J. Proc Natl Acad Sci USA, 1980, 77, 1736–1740.
4. Richmond, T. J. J Mol Biol 1984, 178, 63–89.
5. Hasel, W.; Hendrickson, T. F.; Still, W. C. Tetrahed Comput Methodol 1988, 1, 103–116.
6. Dodd, L. R.; Theodorou, D. N. Mol Phys 1991, 72, 1313–1345.
7. Wesson, L.; Eisenberg, D. Protein Sci 1992, 1, 227–235.
8. Perrot, G.; Cheng, B.; Gibson, K. D.; Vila, J.; Palmer, K. A.; Nayeem, A.; Maigret, B.; Scheraga, H. A. J Comput Chem 1992, 13, 1–11.
9. von Freyberg, B.; Braun, W. J Comput Chem 1993, 14, 510–521.
10. Eisenhaber, F.; Argos, P. J Comput Chem 1993, 14, 1272–1280.
11. Mumenthaler, C.; Braun, W. J Mol Model 1995, 1, 1–10.
12. Kurochkina, N.; Lee, B. Protein Eng 1995, 8, 437–442.
13. Liotard, D. A.; Hawkins, G. D.; Lynch, G. C.; Cramer, C. J.; Truhlar, D. G. J Comput Chem 1995, 16, 422–440.
14. Sridharan, S.; Nicholls, A.; Sharp, K. A. J Comput Chem 1995, 16, 1038–1044.
15. Sanner, M. F.; Olson, A. J.; Spehner, J.-C. Biopolymers 1996, 38, 305–320.
16. Gabdoulline, R. R.; Wade, R. C. J Mol Graphics 1996, 14, 341–353.
17. Cossi, M.; Mennucci, B.; Cammi, R. J Comput Chem 1996, 17, 57–73.
18. Fraczkiewicz, R.; Braun, W. J Comput Chem 1998, 19, 319–333.
19. Cui, Y.; Chen, R. S.; Wong, W. H. Proteins 1998, 31, 247–257.
20. Weiser, J.; Shenkin, P. S.; Still, W. C. J Comput Chem, 1999, 20, 217–230.
21. Mohamadi, F.; Richards, N. G. J.; Guida, W. C.; Liskamp, R.; Lipton, M.; Caufield, C.; Chang, G.; Hendrickson, T.; Still, W. C. J Comput Chem 1990, 11, 440–467. (We used MacroModel Version 6.5.)
22. Jorgensen, W. L.; Tirado–Rives, J. J Am Chem Soc 1988, 110, 1657–1666.
23. Gibson, K. D.; Scheraga, H. A. Mol Phys 1987, 62, 1247–1265.
24. Gogonea, V.; Osawa, E. J Comput Chem 1995, 16, 817–842.
25. Stouten, P. F. W.; Frömmel, C.; Nakamura, H.; Sander, C. Mol Simul 1993, 10, 97–120.
26. Korn, G. A.; Korn, T. M. Mathematical Handbook for Scientists and Engineers; McGraw–Hill: New York, 1968, p. 471.
27. Weiser, J.; Weiser, A. A.; Shenkin, P. S.; Still, W. C. J Comput Chem 1998, 19, 797–808.
28. Bernstein, F. C.; Koetzle, T. F.; Williams, G. J. B.; Meyer, E. F., Jr.; Brice, M. D.; Rodgers, J. R.; Kennard, O.; Shimanouchi, T.; Tasumi, M. J Mol Biol 1977, 112, 535–542. The Brookhaven Protein Data Bank is accessible via the internet at http://pdb.pdb.bnl.gov.
29. Weiser, J.; Holthausen, M. C.; Fitjer, L. J Comput Chem 1997, 18, 1264–1281. Supplementary material including data for sipholenol-A monoacetate is available via the internet at http://journals.wiley.com/0192-8651/wilma/wilma.cgi/v18.1264.html.
30. (a) Weiner, S. J.; Kollman, P. A.; Case, D. A.; Singh, U. C.; Ghio, C.; Alagona, G.; Profeta, S., Jr.; Weiner, P. J Am Chem Soc 1984, 106, 765–784; (b) McDonald, D. Q.; Still, W. C. Tetrahedron Lett. 1992, 33, 7743–7746.
31. Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. J Am Chem Soc 1990, 112, 6127–6129.
32. Ponder, J. W.; Richards, F. M. J Comput Chem 1987, 8, 1016–1024.
33. Polak, E.; Ribiere, G. Rev Franc Inf Recherche Oper 1969, 16-R1, 35.